Efficient processing of acoustic signals for high rate information transmission over sparse underwater channels [⋆]

M.Stojanovic

Massachusetts Institute of Technology
Cmabridge, MA 02139
millitsa@mit.edu

Abstract

For underwater acoustic channels where multipath spread is measured in tens of symbol intervals at high transmission rates, multichannel equalization required for bandwidth-efficient communications may become prohibitively complex for real-time implementation. To reduce computational complexity of signal processing and improve performance of data detection, receiver structures that are matched to the physical channel characteristics are investigates. A decision-feedback equalizer is designed which relies on an adaptive channel estimator to compute its parameters. The channel estimate is reduced in size by selecting only the significant components, whose delay span is often much shorter than the multipath spread of the channel. Optimal coefficient selection (sparsing) is performed by truncation in magnitude. This estimate is used to cancel the post-cursor ISI prior to linear equalization. Spatial diversity gain is achieved by a reduced-complexity pre-combining method which eliminates the need for a separate channel estimator/equalizer for each array element. The advantages of this approach are reduction in the number of receiver parameters, optimal implementation of sparse feedback, and efficient parallel implementation of adaptive algorithms for the pre-combiner, the fractionally-spaced channel estimators and the short feedforward equalizer filters. Receiver algorithm is applied to real data transmitted at 10 kbps over 3 km in shallow water, showing excellent results.

Key words: underwater acoustic communications, adaptive channel estimation, sparse equalization, phase synchronization, reduced-complexity combining.

[⋆] This work was supported in part by the ONR MURI grant N00014-07-1-0738.

1 Introduction

Underwater wireless (acoustic) communications have traditionally relied on noncoherent modulation / detection methods to avoid the problems of long multipath and time-variations encountered in the majority of underwater environments, especially on the horizontal transmission channels. While noncoherent signaling provides robustness to channel distortions, and still represents the preferred method in commercially available acoustic modems, it lacks the bandwidth efficiency necessary for achieving high-rate digital communications over severely band-limited underwater acoustic channels. Bandwidth-efficient underwater acoustic communications can be achieved by employing spatial diversity combining and equalization of PSK or QAM signals. The receiver structure that has been found useful in the majority of present applications is a multichannel decision-feedback equalizer (DFE)[1]. This receiver consists of a bank of adaptive feedforwad filters, one per array element, followed by a decision-feedback filter. It has been implemented in the prototype high-rate acoustic modem developed at the Woods Hole Oceanographic Institution, and shown to perform well in a variety of sea trials [2]. Due to the complex nature of the propagation channel, whose impulse response may extend over several tens or even a hundred milliseconds, causing severe intersymbol interference (ISI) at high transmission rates, the signal processing required by this receiver may become prohibitively complex in certain situations, ultimately limiting the achievable bit rate, as well the receiver's applicability to difficult channels. It is of interest for such situations to develop alternative processing methods that provide the same or similar performance at a lower computational complexity.

Reduction in computational complexity can be achieved in two ways: (1) by using efficient adaptive algorithms, and (2) by altering the conventional receiver structure to obtain one with fewer elements that need to be adjusted adaptively. The work in this area to date has focused on using a class of low-complexity LMS algorithms with improved tracking properties [5] and on several techniques for reducing the size of adaptive filters. In particular, these techniques include (1) reducing the size of the spatial combiner [3]; (2) employing time-reversal to perform adaptive matched filtering prior to equalization [4], and (3) reducing the size of the adaptive equalizer through sparsing [6,7]. The major focus of the present paper is on the last form of complexity reduction. Reduced-complexity spatial combining method of [3] is used with a decision-feedback equalization method that is based on sparse channel estimation. In doing so, our goal is to consider a receiver structure that is matched to the physical characteristics of the propagation channel. In particular, by tracking the channel explicitly, and not implicitly through the coefficients of a large equalizer, it is possible to design an adaptive processing method that takes into account only the significant channel components. The composite time span of these components is often much shorter than the overall multipath spread, leading to the desired reduction in complexity.

Sparse, or tap-selective equalization is a method that has been considered for communications over horizontal underwater acoustic channels [5,6], for broadband wireless radio channels [8,9] and for terrestrial HDTV systems [10]. An *ad hoc* sparse DFE used for equalization of underwater acoustic channels [5] determines the positions of significant equalizer taps by computing the full-size solution initially, but keeping only those taps whose magnitude exceeds a pre-determined threshold. The coefficients of the feedforward and the feedback filter are updated as a single coefficient vector based on minimization of the mean-squared error in data symbol estimation. This tap selection method is not optimal because the input signal to the equalizer is not white. Selection of optimal tap locations in such conditions is a difficult problem involving an exhaustive search. This fact serves as a motivation for developing a channel-estimation based equalizer: the input to the channel estimator is a data sequence of uncorrelated symbols, and, thus, optimal tap selection can be accomplished simply by truncation in magnitude [6].

General methods for estimation of sparse communication channels [11] simultaneously search for the location of significant channel taps and their magnitudes. Casting the channel identification problem into the framework of on-off keying (a channel tap is either on or off), these methods employ techniques such as the Viterbi algorithm or sphere decoding to perform either joint or iterative detection of the channel taps. Their performance was investigated through simulation and analysis, both in terms of the mean squared channel estimation error, and in terms of the bit error rate that results when the channel estimates are used to implement a correlation receiver. The results quantify the benefits of tap-selective solutions over the conventional, full-size ones. A method based on the technique of matching pursuits, specifically modified for identification of sparse underwater acoustic channels, was also recently proposed [7]. In this method, the iterative search for tap locations and magnitudes is extended to include an unknown frequency offset associated with each significant tap. It was applied to real data, showing superior performance in cases of extreme difficulty, at the price of additional computations needed to identify the significant tap locations.

In this paper, we focus on a method for decision-feedback equalization based on adaptive sparse channel estimation, targeting reduction in computational complexity and simplicity of implementation. The equalizer performs a two-step procedure. First, it uses a channel estimate and previous symbol decisions to determine the ISI resulting from the post-cursors. The post-cursor ISI term is subtracted from the input signal, and the so-obtained signal is equalized by a linear feedforward filter. Symbol decisions are then made, which are also used for channel estimation in a decision-directed mode. Thus, the feedback filter is not implemented explicitly, but through the channel estimation feedback. The optimal DFE solution obtained by this method in a stationary case is identical to the one obtained by solving for the equalizer coefficients directly. However, in practice this method has several advantages. Decoupling of channel estimation from equalization allows parallel implementation of efficient adaptive algorithms for the channel estimator and the feedforward equalizer. Sparsing of the channel estimate results in the optimal selection of feedback taps which eliminates the unnecessary noise in data detection, thus improving the receiver performance.

In the majority of underwater communication scenarios, the SNR observed on a single channel is not sufficient for reliable equalizer performance. Hence, it is imperative that some form of multichannel processing be used. Decision-feedback equalization via channel estimation can readily be extended to the multichannel case. In a commonly used multichannel implementation, every element of the receiving array is accompanied by an adaptive feedforward equalizer filter. To avoid the computational demands of such implementation, the front section of the receiver is modified. It uses a spatial pre-combiner, which reduces the number of adaptive filters required per input channel, but preserves the multichannel processing gain [3].

The paper is divided into three parts. The first part, given in Sec.2, outlines the concept of equalization via channel estimation. For simplicity, the discussion is limited to the single-channel case. Adaptive implementations are discussed and a computationally efficient channel estimator is proposed. The problem of tap selection is addressed, and illustrated through a numerical example. The second part, given in Sec.3, is devoted to the multichannel receiver. This section includes receiver parameter optimization based on a minimum mean squared error (MMSE) criterion, and a complete algorithm for adaptive implementation of the pre-combiner and the equalizer. The algorithm also incorporates a multichannel decision-directed carrier recovery scheme. The third part, given in Sec.4, contains the results of experimental data processing. Receiver performance is demonstrated on real data transmitted using QPSK at 10 kilobits per second over 3 km in shallow water, showing excellent results. The conclusions are summarized in Sec.5.

2 Channel Estimation Based Equalization

The method for determining the equalizer coefficients from a channel estimate is based on an alternative interpretation of the classical MMSE DFE.

2.1 DFE: an alternative interpretation

Let the input signal to the equalizer be the phase-synchronous baseband signal, coarsely aligned in time:

$$v(t) = \sum_n d(n)h(t - nT) + w(t) \tag{1}$$

where $d(n)$ is an i.i.d. sequence of unit-variance data symbols transmitted at times nT; $h(t)$ is the overall channel response, including transmitter and receiver filtering, and $w(t)$ is the additive noise. This signal is sampled at the Nyqist or higher rate. Without loss of generality, we assume a sampling rate of $2/T$ for a signal band-limited to $1/T$. The signal samples are arranged in a column vector:

$$\begin{bmatrix} v(nT + M_1T/2) \\ \vdots \\ v(nT + T/2) \\ v(nT) \\ v(nT - T/2) \\ \vdots \\ v(nT - M_2T/2) \end{bmatrix} = \sum_k \begin{bmatrix} h(kT + M_1T/2) \\ \vdots \\ h(kT + T/2) \\ h(kT) \\ h(kT - T/2) \\ \vdots \\ h(kT - M_2T/2) \end{bmatrix} d(n-k) + \begin{bmatrix} w(nT + M_1T/2) \\ \vdots \\ w(nT + T/2) \\ w(nT) \\ w(nT - T/2) \\ \vdots \\ w(nT - M_2T/2) \end{bmatrix} \tag{2}$$

or shortly,

$$\mathbf{v}(n) = \sum_k \mathbf{h}(k)d(n-k) + \mathbf{w}(n) \tag{3}$$

The reference channel vector $\mathbf{h}(0)$ has a time span $(-M_2T/2, M_1T/2)$ which is chosen to capture all of the channel response $h(t)$. Coarse alignment is normally performed such that the magnitude of cross-correlation between the sequences $v(nT + lT/2)$ and $d(n)$ is maximal for $l = 0$, in which case the reference element $h(0)$ is the channel coefficient with maximal amplitude.

The DFE with feedforward filter coefficients a_k arranged in a vector $\mathbf{a}$, and feedback coefficients b_k, estimates the data symbol as

$$\hat{d}(n) = \mathbf{a}'\mathbf{v}(n) - \sum_{k>0} b_k^* \tilde{d}(n-k) \tag{4}$$

where $\tilde{d}(n)$ is the correct data symbol in the training mode or the symbol decision in the decision-directed mode. Prime denotes conjugate transpose, and all the vectors are defined as column vectors. In the absence of decision errors, the optimal choice of the feedback taps is the one for which post-cursor ISI is completely canceled:

$$b_k^* = \mathbf{a}'\mathbf{h}(\mathbf{k}),\ k > 0 \tag{5}$$

For this choice, the decision variable is expressed as

$$\hat{d}(n) = \mathbf{a}'[\sum_{k\le 0}\mathbf{h}(k)d(n-k) + \mathbf{w}(n)] = \mathbf{a}'\mathbf{v}_f(n) \tag{6}$$

This expression is used to determine the MMSE solution for the feedforward filter in terms of the channel vector shifts $\mathbf{h}(k)$ and the noise covariance $\mathbf{N} = E\{\mathbf{w}(n)\mathbf{w}'(n)\}$. The solution is given by

$$\mathbf{a} = \mathbf{R}_f^{-1}\mathbf{h}(0) \tag{7}$$

where $\mathbf{R}_f$ is the covariance of the term $\mathbf{v}_f(n)$, and, for independent data symbols, it is given by

$$\mathbf{R}_f = E\{\mathbf{v}_f(n)\mathbf{v}_f'(n)\} = \sum_{k\le 0}\mathbf{h}(k)\mathbf{h}'(k) + \mathbf{N} \tag{8}$$

Expressions (8), (7) and (5) define the classical MMSE DFE. They also provide insight into the needed sizes of equalizer filters.

A usual approach to implementing an adaptive DFE is to group the feedforward and the feedback filter taps in a composite vector, and apply a least-squares algorithm to compute this vector recursively from the input signal vector $\mathbf{v}(n)$ and the previous decisions $\tilde{d}(n-k),\ k > 0$, using the data estimation error $e(n) = d(n) - \hat{d}(n)$. The so-obtained equalizer taps are used to filter the received signal, and subtract the post-cursors ISI term according to the expression (4).

An alternative implementation is based on the expression (6). For this implementation, the equivalent feedforward signal $\mathbf{v}_f(n)$ has to be obtained first. However, this signal cannot be measured directly. Instead, it can be reconstructed. Reconstruction is based on the fact that this signal can be represented as

$$\mathbf{v}_f(n) = \mathbf{v}(n) - \mathbf{v}_b(n) \tag{9}$$

where

$$\mathbf{v}_b(n) = \sum_{k>0} \mathbf{h}(k) d(n-k) \tag{10}$$

is the equivalent feedback signal that can be obtained from the previous decisions and a channel estimate. Hence, the modified DFE implementation is defined as follows:

(1) Using a channel estimate $\hat{\mathbf{h}}(0)$, its shifts $\hat{\mathbf{h}}(k)$ for $k > 0$, and previous decisions, determine the equivalent feedforward input signal as

$$\hat{\mathbf{v}}_f(n) = \mathbf{v}(n) - \sum_{k>0} \hat{\mathbf{h}}(k) \tilde{d}(n-k) = \mathbf{v}(n) - \hat{\mathbf{v}}_b(n) \tag{11}$$

(2) Apply an adaptive linear equalizer to this signal to obtain the data symbol estimate

$$\hat{d}(n) = \mathbf{a}'(n) \hat{\mathbf{v}}_f(n) \tag{12}$$

Any adaptation algorithm may be considered.

In this approach, the feedback filter taps b_k are not computed explicitly at all. The feedforward filter operates on the equivalent signal, from which the post-cursor ISI has been removed. In this sense, adaptive feedforward filtering is different from the original approach in which the filter operates directly on the received signal.

A feature worth noting is that the estimate of the feedback signal, $\hat{\mathbf{v}}_b(n)$, obeys a shifting law:

$$\hat{\mathbf{v}}_b(n) =\downarrow \hat{\mathbf{v}}_b(n-1) + \hat{\mathbf{h}}(1) \tilde{d}(n-1) \tag{13}$$

where ↓ indicates shifting of the vector downward by two elements (in general, by as many elements as there are samples per one symbol interval) and filling the top by zeros. This property eliminates the need to carry out the entire summation for determining the post-cursor ISI every time a new data decision becomes available. It was shown originally in Ref. [5] for a T-spaced equalizer.

2.2 Adaptive channel estimator and equalizer implementation

There are many possibilities for implementing the channel estimator and the equalizer adaptively. Each should be chosen according to the channel at hand. Some scenarios are the following:

1. Fixed channel estimate / adaptive equalizer.
Channel is estimated from the packet preamble, and this estimate is frozen for the duration of the packet. Short feedforward equalizer is adapted throughout the packet. This approach is suitable for high speed radio receivers, where computational complexity is of paramount importance, but the channel can safely be assumed constant over the packet duration [8,9].

2. Adaptive channel estimate / adaptive equalizer.
Channel estimator and a short equalizer are updated throughout the packet. Channel estimation is independent of equalization, except that it relies on symbol decisions in the decision-directed mode. This approach is the first choice for underwater acoustic communications that use packets long enough to support significant channel changes. The receiver structure also allows arbitrary choice of updating intervals.

The equalizer and the channel estimator are updated separately. Thus, they may use different adaptive algorithms. For channel estimation, a number of adaptive algorithms can be devised based on the modeling equation (1). For instance, the estimates of channel coefficients $h(nT)$ can be obtained from the coefficients of an adaptive filter that uses the data sequence $d(n)$ as its input to estimate the signal $v(nT)$. For a fractional spacing of $T/2$, two estimators are needed to generate a $T/2$ spaced channel response. Since computational complexity is the focal point of this work, and the channel responses are expected to be of considerable length as measured in the number of samples $M = M_1 + M_2 + 1$, it is of interest to use a computationally simple adaptive algorithm for channel estimation. The algorithm proposed below is based on the fact that the input signal can be modeled as in the expression (3). From this expression it follows that

$$\mathbf{h}(0) = E\{\mathbf{v}(n)d^*(n)\} \tag{14}$$

To obtain an estimate of the above vector, a simple stochastic approximation can be used:

$$\hat{\mathbf{h}}[n] = (1-\lambda_{ch})\sum_{i=0}^{n}\lambda_{ch}^{n-i}\mathbf{v}(i)d^*(i) \tag{15}$$

where $\hat{\mathbf{h}}[n]$ denotes the estimate of $\mathbf{h}(0)$ obtained in the nth iteration, i.e., $\hat{\mathbf{h}}[n] = \hat{\mathbf{h}}(0,n)$, and λ_{ch} is an exponential forgetting factor. The scaling factor $(1-\lambda_{ch})$ ensures an asymptotically unbiased estimate. This expression gives way to a very simple recursion:

$$\hat{\mathbf{h}}[n] = \lambda_{ch}\hat{\mathbf{h}}[n-1] + (1-\lambda_{ch})\mathbf{v}(n)d^*(n) \tag{16}$$

The estimate $\hat{\mathbf{h}}[n]$ is chosen to span all of the significant channel response, but it suffices to keep only the channel coefficients with significant amplitude. In this regard, a key feature of the algorithm is that the channel coefficients are updated independently. While such a method is not optimal if there is correlation between the channel coefficients, it allows fast tracking of the channel response.

Note that there is a choice when it comes to channel estimation: to update the entire channel estimate vector, or to update only the selected taps. Updating the entire vector provides constant on-line monitoring of the time-variations in the channel (even though all of the taps are not used for post-cursor ISI suppression). This approach may be useful if tap migration is present due to residual Doppler effect (note that this may be different for different taps [7]). Updating only the selected taps provides reduction in computational complexity. The choice should be made based on the properties of a particular channel, and the desired performance. The simplicity of the algorithm presented above serves to somewhat reconcile the two requirements.

The adaptive algorithm for the equalizer is commonly chosen from the LMS or the RLS family. A least-squares algorithm for the equalizer vector $\mathbf{a}$ will operate on the equivalent input signal $\mathbf{v}_f(n)$, driven by the error $e(n)$. There is also a possibility that the feedforward equalizer be calculated from the channel estimate, making use of the relation (7). The matrix inverse in this case would be computed at the beginning of a data packet, and possibly frozen for the packet duration on the grounds that the channel covariance changes more slowly than the channel realization. However, the advantage of updating the equalizer independently is that it may compensate to some degree for the channel estimation errors.

2.3 Tap selection

Decomposing the decision-feedback equalization into the channel estimation part and the feedforward equalization part eases the difficult problem of optimal DFE tap selection. Because the feedback is implemented a priori through the channel estimate, feedback tap selection is obviously made by choosing only the significant coefficients of the channel estimate. As long as the input to the channel estimator is a data sequence of uncorrelated symbols, optimal feedback tap selection is performed by choosing only the channel estimate coefficients whose amplitude is above some threshold. In addition, this choice can be made adaptively throughout the packet. (Note the difference between sparsing the channel estimate and sparsing the coefficients of the equalizer.)

Selection of the best feedforward filter taps, however, remains a difficult optimization problem. In general, not all the M coefficients will be updated in the equalizer vector, as it is desired to use only a short feedforward filter with $N < M$ non-zero taps. The problem of feedforward equalizer tap selection is investigated below. Three approaches come to mind:

1. Optimally sparsed filter.
The search is conducted among all the sparsing patterns $\mathbf{S}$ to find the one for which the minimum mean squared data estimation error obtained with the filter of given size N is minimal:

$$E_{min}(N) = \min_{\mathbf{S}_{M\times N}} \{1 - \mathbf{h}'(0)\mathbf{S}[\mathbf{S}'\mathbf{R}_f\mathbf{S}]^{-1}\mathbf{S}'\mathbf{h}(0)\} \tag{17}$$

The selection matrix $\mathbf{S}$ is of size $M \times N$. Each column contains exactly one 1 among all zeros. Its meaning is that of selecting the elements of the vector $\hat{\mathbf{v}}_f(n)$ to form an input signal $\mathbf{v}_f^{(s)}(n) = \mathbf{S}'\hat{\mathbf{v}}_f(n)$ for the sparse feedforward filter with $N \leq M$ non-zero taps.

Note from the expression (17) that tap selection according to magnitude would be optimal only if $\mathbf{R}_f \sim \mathbf{I}$, i.e. for uncorrelated input signal. However, this is not the case in the equalization problem. Finding the optimal sparsing pattern in general involves an exhaustive search whose complexity may be prohibitive for a practical implementation.

2. Approximation of the optimization criterion to allow for a more efficient search.
One such approximation is investigated in Ref.[6]. In this reference, a T-spaced feedforward filter is considered and the approximation is made by searching among all the single-tap filters and selecting those N which result in the lowest MMSEs. In other words, only the M selection vectors $\mathbf{s}$ of size $M \times 1$ are considered. Each vector contains a single 1 at a different location, and results in the MMSE

$$E(1) = 1 - \mathbf{h}'(0)\mathbf{s}[\mathbf{s}'\mathbf{R}_f\mathbf{s}]^{-1}\mathbf{s}'\mathbf{h}(0) \tag{18}$$

The M values of this single-tap MMSE are computed, and the N lowest ones are identified. The corresponding taps are chosen to form the equalizer vector.
3. An ad hoc method.

For example, a feedforward filter can be chosen with contiguous taps, but of total length much shorter than the length of the channel estimate. (Ideally, the number of contiguously spaced feedforward taps will be chosen at least equal to the number of channel taps.) In reference to the channel vector, the feedforward filter coefficients of the DFE are arranged as $\mathbf{a}' = [0 \ldots 0 \, a(-N_1) \ldots \, a(N_2) \, 0 \ldots 0]^*$. The channels of interest are characterized by large M, and in particular by large M_1, the part of the response responsible for post-cursor ISI. In this case, it will be of interest to use a short feedforward filter with $N_1 < M_1$ taps. This approximation is justified by the fact that post-cursor ISI has been canceled from the feedforward filter input. It seems to be effective for channels with decaying multipath intensity profile as long as N is large enough to capture sufficient signal energy. Extending the feedforward span N_1 improves on the matched filtering, but increases the irreducible pre-cursor ISI.

Another ad hoc method would be to assign feedforward filter taps to match the significant taps of the channel estimate $\hat{\mathbf{h}}(0)$. However, its effectiveness is not obvious except in special cases of distinctly sparse channels with comparable energy of multipath arrivals.

2.4 Example

To demonstrate the design concepts, and to analyze the various tap selection methods, a numerical example is constructed. A two-path time-invariant channel is considered, with path amplitude ratio 2/1 and relative path delay of 4.25 symbols. The transmitter filter is a spectral raised cosine with roll-off factor 0.25 and truncation length of ±4 symbol intervals. (This is the actual filter response used to generate experimental signals of Sec.4.) The channel introduces additive white Gaussian noise (AWGN). The SNR is 20 dB. The channel estimator looks 2 symbols to the left (M_2=4 taps) and 7 symbols to the right (M_1=14 taps) of the reference tap. The feedforward filter uses 3 taps around the main arrival (N_1=1, N_2=1) and an RLS algorithm. The DFE performance is shown in Fig.1. The figure shows the estimated MSE, the output scatter plot, the tap magnitudes of the feedforward filter, the true channel response (dotted) and the sparse channel estimate (solid). The receiver is trained during the first 38 data symbols, and then switched into the decision-directed mode. There are no decision errors. The channel taps whose magnitude is less than 1/6 of the main tap magnitude are set to zero. In this manner, out of the total of 19 taps, 5 are kept to be used for post-cursor ISI cancellation. An ad hoc choice of few taps around the main arrival works very well in this type of channel.

The problem of optimal tap allocation is analyzed in Fig.2. This figure shows the theoretical results for the sparse MMSE DFE. The channel is truly sparse, as shown in the upper right corner. The total channel length is set to $M = 15$. Shown in the upper left corner is the MMSE versus the number of equalizer taps which are selected optimally (tap selection according to rule 1 of Sec.2.3). Naturally, the MMSE decreases as the total number of taps increases, and the improvement obtained after a certain size becomes negligible. It is interesting to compare the magnitudes of the optimally chosen feedforward filter taps to

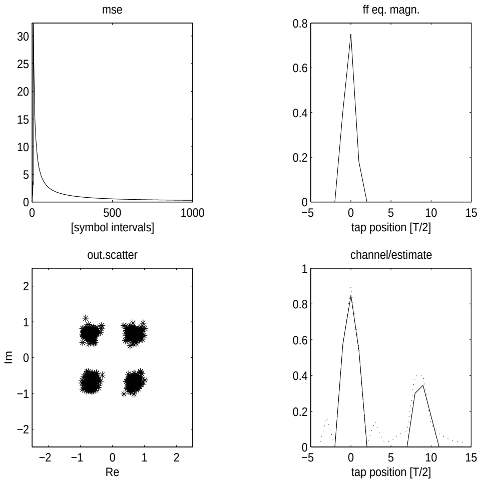

Fig. 1. Performance of the channel-estimation-based DFE.ad hoc sparsing of the feedforward filter. $N = 3, M = 19, N_t = 38, G_0 = 1/6, \lambda_{eq} = 0.999, \lambda_{ch} = 0.99$, SNR=20 dB.

the channel response magnitude. The feedforward equalizer tap magnitudes are shown below the channel response for several total equalizer lengths N. The equalizer size $N = 5$ is the first whose optimal selection includes taps outside of the main arrival region. Although as N increases, the optimally selected taps tend to include the region of the second arrival, they are not limited to it. Tap selection according to the approximate rule proposed in Ref. [6] (rule 2 of Sec.2.3) is illustrated in the lower left corner. Interestingly, if this rule were applied to determine the best 5 taps, those taps would correspond to the nonzero elements of the channel vector (taps 2, 3, 4, 11 and 12, as numbered in the figure). Application of the optimal rule, however, gives a different solution in which more taps are allocated to the region of the main arrival (taps 2, 3, 4, 5, and 12). These observations stress the difficulty of choosing the tap selection criterion for a practical implementation. However, the fact that both rules contain taps in the region of the main arrival provides an encouragement for choosing the taps in an ad hoc manner (criterion 3 of Sec.2.3). This rule, which requires no computations, is used in the experimental data processing described in Sec.4.

While the number of feedforward taps is determined in advance and their location chosen according to one of the rules described, the sparse feedback is implemented by truncating the channel estimate. The choice of truncation threshold obviously influences the receiver performance. This effect is illustrated in Fig.3 which shows the output SNR as a function of the truncation threshold. Simulation results demonstrate the existence of an optimal threshold for which the output SNR is maximized. Corresponding to the optimal threshold is the number of channel coefficients that are kept for evaluating the post-cursor ISI. As expected, this number is less than the total number of coefficients used to represent the channel response. At threshold 0, all 19

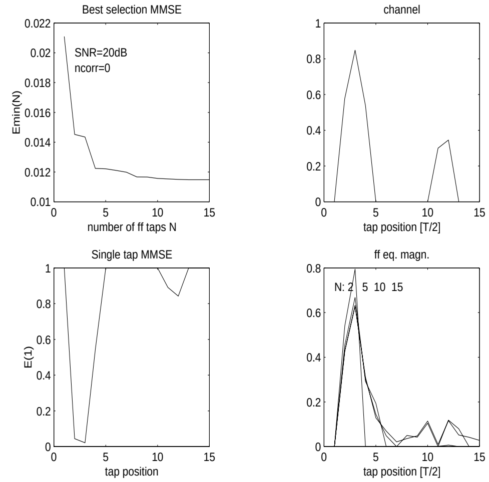

Fig. 2. Theoretical performance of the optimally sparsed DFE. “Best selection MMSE” is the value of $E_{min}(N)$, the MMSE obtained with the optimally selected positions of N feedforward taps. SNR of 20 dB and uncorrelated noise are assumed for evaluating the MMSE. The channel is truly sparse, and shown in the upper right corner. The optimal feedforward tap magnitudes for several sizes N are shown in the lower right corner. Lower left corner shows the MSE obtained by a single tap equalizer, which is used in tap selection based on an approximation of the optimal criterion. The best choice in this case is tap # 3 (this choice results in $E(1) = E_{min}(1)$). The second best is tap #2, then 4, 12, 11, etc.

coefficients are used. The number of coefficients kept at threshold 0.1 is 5-6 (exact value depends on the noise realization), and decreases to 2-3 at threshold 0.5. After the threshold increases above 0.6, only the strongest coefficient is kept in the channel estimate. The value of the optimal threshold obviously depends on the particular channel response and the input SNR. While this dependence may be difficult to evaluate analytically, a threshold close to optimal can easily be found in practice by tuning the receiver while monitoring the average squared error during training.

3 Multichannel Receiver

In the previous section, the principles of channel-estimation-based DFE were described and demonstrated on a simple example generated by computer simulation. In order to apply these principles to real data, however, a multichannel processing gain is required in the majority of underwater acoustic communication scenarios. Hence, the sparse DFE needs to be cast in the framework of a multichannel receiver.

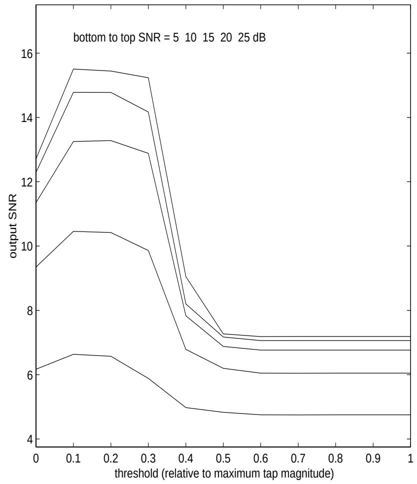

Fig. 3. SNR at the equalizer output as a function of the threshold used for sparsing the channel estimate. The input SNR is determined by the AWGN power. Threshold is given relative to the absolute value of the maximum channel coefficient: a threshold of 0 indicates that all coefficients are kept in the channel estimate used for post-cursor ISI calculation, while a threshold of 1 indicates that only one (the strongest, or the reference coefficient) is kept. The feedforward equalizer uses 3 contiguously spaced taps around the reference tap; the channel estimate has 19 coefficients. The curves are obtained by averaging 500 independent simulation runs.

3.1 Receiver structure

A multichannel receiver in its conventional form consists of a number of input channels, K, to each of which an adaptive feedforward filter is assigned. This receiver structure is effective for various types of underwater acoustic channels; however, for a large number of input channels, its computational complexity becomes high, because each of the K channels requires an adaptive filter of length that is often measured in tens of coefficients. To alleviate this problem, a computationally efficient LMS algorithm was considered in Ref. [2] as an alternative to a fast RLS, but the inevitable trade-off in convergence rate was found to limit its usefulness.

Further reduction in complexity is possible by reducing the size of the front spatio-temporal processor [3]. In this approach, the K spatially distributed input channels are first combined into a smaller number of channels, P. No temporal processing (filtering) is used in doing so, but only weighted combining using adaptively determined weights. The resulting P channels are then equalized. The approach of pre-combining, or reduced-complexity combining has proven to be very effective in processing different types of real data. What is interesting to bear in mind is that a small value of P, usually only two or three channels (but rarely one) is sufficient for extracting the multichannel processing gain. This property stems from the broad-band nature of underwater acoustic communication signals. The reduced-complexity multichannel structure is found to be appropriate for use with sparse, channel estimation aided DFE, for reasons that will become apparent shortly. The resulting receiver structure is shown in Fig.4.

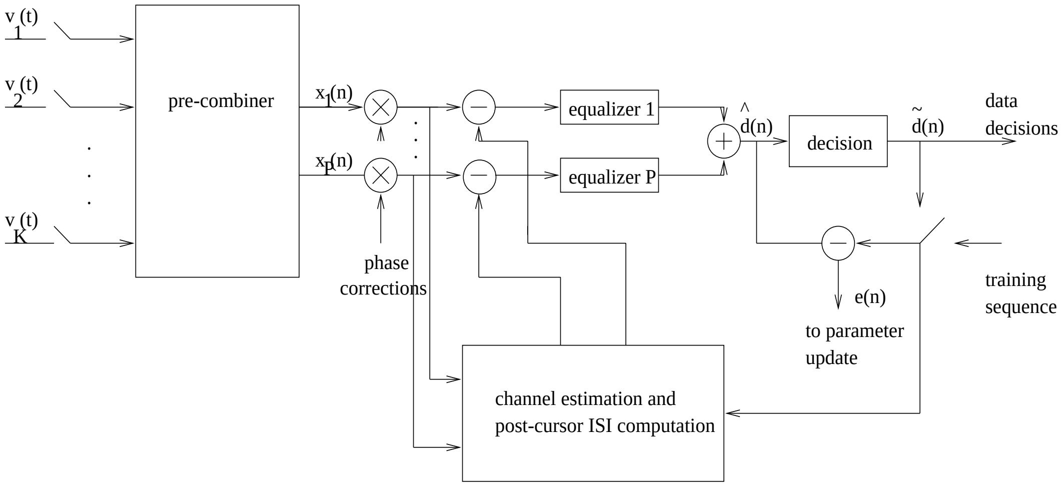

Fig. 4. Reduced complexity multichannel DFE incorporates a K to P pre-combiner and a P-channel DFE. The DFE is based on channel estimation and sparsing.

In addition to multichannel combining, a practical receiver must incorporate phase tracking. Specifically, let the input signals be given by

$$v_k(t) = \sum_n d(n) h_k(t - nT) e^{j\phi_k(t)} + w_k(t), \; k = 1, \dots K \tag{19}$$

where, as before, $d(n)$ denotes the transmitted data sequence and $w_k(t)$ is the additive noise observed at the kth receiving element, but the phase deviation is included explicitly as $\phi_k(t)$, rather than being incorporated into the (complex-valued) channel response $h_k(t)$. This is done so as to separate the rapid time-variation of the phase from that of the channel which usually varies much more slowly. While the rapid variation must be tracked by a phase-locked loop (PLL), the slow one can be handled by the channel estimator. There are several choices for implementing a decision-directed PLL. Phase tracking can be performed before combining, i.e., individually for all K channels; after pre-combining, i.e., using only P phase estimates, or after linear equalization using a single phase estimate. In what follows, the second choice will be used, on the grounds that the K phases $\phi_k(t)$ are usually correlated, but the P signals obtained after pre-combining may contain independently varying components. If this is not the case, and there is strong correlation between the P phase estimates, modification of the phase locked loop from P channels to a single channel is straightforward. Since P is usually chosen to be small, there is not much to be gained in complexity reduction by eliminating these additional phase estimates.

The question that arises in the multichannel receiver design is how many channel estimates are needed, or, equivalently, at which point in the receiver should the post-cursor ISI be removed. If the channel responses $h_k(t)$ from the transmitter to each of the receiving elements were known, this could be done directly on the K input signals $v_k(t)$. However, we already know that a single input signal does not yield sufficient SNR for data detection (hence multichannel processing), and consequently, it is not suitable for individual channel estimation either. Thus, a processing gain must be extracted *before* channel estimation is attempted. The approach of pre-combining is precisely suited for this purpose. As shown in Fig.4, the channel estimates are formed after extracting the pre-combining gain.

3.2 Parameter optimization

Receiver parameters that need to be determined include the $K \times P$ pre-combiner weights, arranged in a matrix $\mathbf{C}$; the feedforward filter coefficients arranged in P vectors denoted by $\mathbf{a}_p$; the P phase estimates $\hat{\theta}_p$, and the P channel estimates denoted by $\hat{\mathbf{f}}_p,\ p = 1, \ldots P$. These parameters are optimized jointly in a manner that minimizes the MSE in data detection.

The signals obtained after pre-combining are given by

$$x_p(t) = \sum_{k=1}^{K} c^*_{p,k} v_k(t) = \mathbf{c}'_p \mathbf{v}(t),\ p = 1, \ldots P \tag{20}$$

where $\mathbf{c}_p$ is the pth column of the pre-combining matrix $\mathbf{C}$. The signals are sampled once every T_s seconds. (As before, we may assume $T_s = T/2$.) Let $\mathbf{V}(n)$ denote the matrix of all signal samples used for processing at time instant nT, associated with detection of the nth data symbol $d(n)$:

$$\mathbf{V}(n) = \begin{bmatrix} v_1(nT + M_1T_s) & \cdots & v_1(nT) & \cdots & v_1(nT - M_2T_s) \\ \vdots & & \vdots & & \vdots \\ v_K(nT + M_1T_s) & \cdots & v_K(nT) & \cdots & v_K(nT - M_2T_s) \end{bmatrix} \tag{21}$$

The signals at the output of the pre-combiner can now be arranged in a matrix

$$\mathbf{X}(n) = \mathbf{C}'\mathbf{V}(n) = \begin{bmatrix} x_1(nT+M_1T_s) & \cdots & x_1(nT) & \cdots & x_1(nT-M_2T_s) \\ \vdots & & \vdots & & \vdots \\ x_P(nT+M_1T_s) & \cdots & x_P(nT) & \cdots & x_P(nT-M_2T_s) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{x}_1^T(n) \\ \cdots \\ \mathbf{x}_P^T(n) \end{bmatrix} \tag{22}$$

The signal $\mathbf{x}_p(n)$ is modeled as

$$\mathbf{x}_p(n) = \sum_i \mathbf{f}_p(i)d(n-i)e^{j\theta_p(n)} + \mathbf{z}_p(n) \tag{23}$$

where

$$\mathbf{f}_p(i) = \begin{bmatrix} f_p(iT+M_1T_s) \\ \vdots \\ f_p(iT) \\ \vdots \\ f_p(iT-M_2T_s) \end{bmatrix} \tag{24}$$

represents the i-shift of the vector $\mathbf{f}_p(0)$ which is centered so as to capture all of the channel response (as seen at this point in the receiver); $\theta_p(n)$ represents the phase deviation, and $\mathbf{z}_p(n)$ represents the additive noise. It is the set of channel responses $\mathbf{f}_p(0),\ p = 1, \dots P$ that will be estimated and used for post-cursor ISI suppression, which is thus performed after pre-combining.

Phase correction is performed using the estimates $\hat{\theta}_p$ of the phases θ_p. This operation yields the signals

$$\mathbf{x}_{p\theta}(n) = \mathbf{x}_p(n)e^{-j\hat{\theta}_p(n)},\ p = 1, \dots P \tag{25}$$

which are the inputs to the P-channel DFE.

The analysis of the multichannel DFE that uses channel estimates $\hat{\mathbf{f}}_p(i)$ closely follows that of a single channel DFE given in Sec.2.1 In a conventional multichannel DFE representation, with feedback taps $\{b_i\}, i > 0$, the data symbol estimate is obtained as

$$\hat{d}(n) = \sum_{p=1}^{P} \mathbf{a}_p' \mathbf{x}_{p\theta}(n) - \sum_{i>0} b_i^* \tilde{d}(n-i) \tag{26}$$

Substituting for the signals $\mathbf{x}_{p\theta}(n)$, and assuming for the moment that the phase estimates are correct, one obtains the condition for perfect post-cursor ISI cancellation:

$$b_i^* = \sum_{p=1}^{P} \mathbf{a}_p' \mathbf{f}_p(i), i > 0 \tag{27}$$

Using the channel estimates for lack of true values in the above expressions, the data symbol estimate is computed as

$$\hat{d}(n) = \sum_{p=1}^{P} \mathbf{a}_p' [\mathbf{x}_{p\theta}(n) - \sum_{i>0} \hat{\mathbf{f}}_p(i) \tilde{d}(n-i)] \tag{28}$$

In the last expression, we recognize the backward signal

$$\mathbf{x}_{pb}(n) = \sum_{i>0} \hat{\mathbf{f}}_p(i) \tilde{d}(n-i) \tag{29}$$

and the forward signal, i.e., the input to the pth feedforward filter

$$\mathbf{x}_{pf}(n) = \mathbf{x}_{p\theta}(n) - \mathbf{x}_{pb}(n) \tag{30}$$

The estimation error is given by

$$e(n) = d(n) - \hat{d}(n) \tag{31}$$

This error is used to obtain the MMSE solution for the receiver parameters. In a practical implementation, the MMSE solution is computed recursively to suit the time-varying nature of the channel. The resulting adaptive algorithms for the pre-combiner, the equalizers, the PLL and the channel estimates are given below.

The pre-combiner. To arrive at the solution for the pre-combiner weights, we use the fact that

$$\mathbf{x}_p(n) = \mathbf{V}^T(n) \mathbf{c}_p^* e^{-j\hat{\theta}_p(n)} \tag{32}$$

The data estimate $\hat{d}(n)$ is then written as

$$\begin{aligned}\hat{d}(n) &= [\mathbf{c}_1' \ldots \mathbf{c}_P'] \begin{bmatrix} \mathbf{V}(n)\mathbf{a}_1^* e^{-j\hat{\theta}_1(n)} \\ \vdots \\ \mathbf{V}(n)\mathbf{a}_P^* e^{-j\hat{\theta}_P(n)} \end{bmatrix} - [\mathbf{a}_1' \ldots \mathbf{a}_P'] \begin{bmatrix} \mathbf{x}_{1b}(n) \\ \vdots \\ \mathbf{x}_{Pb}(n) \end{bmatrix} \\ &= \mathbf{c}'\mathbf{y}(n) - \mathbf{a}'\mathbf{x}_b(n) \end{aligned} \tag{33}$$

where we have defined the composite vectors **c** and **a** of the pre-combiner and the equalizers, respectively, as well as the *equivalent* pre-combiner input signal $\mathbf{y}(n)$. Having expressed the estimated quantity as an inner product of the coefficient vector to be determined and the equivalent input signal (plus a fixed term that does not influence the optimization), the MMSE solution for the pre-combiner coefficients is obtained as the usual Wiener filter. The coefficients can be obtained recursively as

$$\mathbf{c}(n+1) = \mathbf{c}(n) + A_1[\mathbf{y}(n), e(n)] \tag{34}$$

where $A_1[\mathbf{u}(n), e(n)]$ denotes an adaptive algorithm A_1, such as LMS or RLS, that operates on the input signal vector $\mathbf{y}(n)$ and the estimation error $e(n)$. For example, for $K = 8$ and $P = 2$, which are the values that will be used in the following section when real data processing is described, there is a total of 16 pre-combiner coefficients. This is a value small enough to permit even the use of a standard RLS algorithm.

The equalizer. To obtain a recursion for updating the equalizer coefficients, the data estimate is expressed as

$$\hat{d}(n) = [\mathbf{a}_1' \ldots \mathbf{a}_P'] \begin{bmatrix} \mathbf{x}_{1f}(n) \\ \vdots \\ \mathbf{x}_{Pf}(n) \end{bmatrix} = \mathbf{a}'\mathbf{x}_f(n) \tag{35}$$

The composite feedforward filter vector is updated as

$$\mathbf{a}(n+1) = \mathbf{a}(n) + A_2[\mathbf{x}_f(n), e(n)] \tag{36}$$

where A_2 is an adaptive algorithm not necessarily the same as A_1. Regarding the equalizer size, a word of caution is in order. Although the vectors $\mathbf{a}_p$ in our treatment so far have been assumed to be of size M, only $N \leq M$ of these elements are actually used. The others are set to zero, and, thus, do not need to be updated. The choice of the N elements has been addressed in Sec.2.3 for the single channel case. In the multichannel case, of course, all the channels do not need to have the same selection of N taps, but the same general principles apply. In the experimental data processing described in the following section, all the equalizers will use an ad hoc method, with the same N contiguous taps selected around the reference channel tap.

The PLL. To obtain the algorithm for updating the phase estimates $\hat{\theta}_p$, the data symbol estimate is written as

$$\hat{d}(n) = \sum_{p=1}^{P} \alpha_p(n) e^{-j\hat{\theta}_p(n)} - \sum_{p=1}^{P} \alpha_{pb}(n) \tag{37}$$

where

$$\alpha_p(n) = \mathbf{a}_p' \mathbf{x}_p(n) \tag{38}$$

$$\alpha_{pb}(n) = \mathbf{a}_p' \mathbf{x}_{pb}(n) \tag{39}$$

The phase estimate is chosen to minimize the MSE in data estimation, and is computed recursively using a second order stochastic gradient approximation. The instantaneous gradient of the MSE with respect to the pth phase estimate is given by

$$\frac{\partial |e^2(n)|}{\partial \hat{\theta}_p} = 2Re\{\frac{\partial e(n)}{\partial \hat{\theta}_p} e^*(n)\} = -2Im\{\alpha_p(n) e^{-j\hat{\theta}_p} e^*(n)\} \tag{40}$$

From this expression follows the second order update:

$$\begin{aligned} \psi_p(n) &= Im\{\alpha_p(n) e^{-j\hat{\theta}_p(n)} e^*(n)\} \\ \hat{\theta}_p(n+1) &= \hat{\theta}_p(n) + K_{f1}\psi_p(n) + K_{f2} \sum_{i=0}^{n} \psi_p(i) \end{aligned} \tag{41}$$

where K_{f1} is the proportional tracking constant, and K_{f2} is the integral tracking constant, often chosen as $K_{f2} = K_{f1}/10$.

The channel estimator. At last, it remains to specify the channel estimation algorithm. The estimate of $\mathbf{f}_p(0)$, generated at the nth iteration is denoted as $\hat{\mathbf{f}}_p(0,n) = \hat{\mathbf{f}}_p[n]$. The estimates of vectors $\hat{\mathbf{f}}_p(1,n)$, needed to determine the post-cursor ISI, are computed by shifting the vectors $\hat{\mathbf{f}}_p[n]$. Since the model (23) implies that

$$\mathbf{f}_p(0) = E\{\mathbf{x}_p(n) e^{-j\theta_p(n)} d^*(n)\},\ p = 1, \dots P \tag{42}$$

the channel estimation algorithm described in Sec.2.2 is applicable to the multichannel case as well. The only difference is that the phase correction needs to be taken into account. Thus, the phase-corrected signals are used for channel estimation:

$$\hat{\mathbf{f}}_p[n] = \lambda_{ch}\hat{\mathbf{f}}_p[n-1] + (1-\lambda_{ch})\mathbf{x}_{p\theta}(n)d^*(n),\ p = 1 \ldots P \tag{43}$$

Each of the estimated vectors $\hat{\mathbf{f}}_p[n]$ is of length M, which may be large if it is to span all of the channel response. However, not all of the M coefficients are used for post-cursor ISI estimation. Sparsing of the channel estimates is performed in an optimal manner by truncation.

A rapidly varying channel necessitates tracking of more than just the selected taps. An underwater channel with a moving transmitter/receiver is such an example. Changes in the propagation path length cause drifting of the channel taps which can be captured by constant monitoring of all the taps. Alternatively, the position of the strongest channel tap could be controlled by performing explicit bit synchronization, i.e., by incorporating a delay-locked loop (DLL) into the receiver. Once the locations of significant taps do not change with time, adaptive channel estimation can be confined to those taps only. For the present application, fractional spacing of the equalizer suffices to extract the correct bit timing.

Finally, it has to be emphasized that the post-cursor ISI term $\mathbf{x}_{pb}(n)$ is given by (29) in terms of a sum that involves all the positive shifts of the estimated channel vector. However, the summation does not need to be carried out explicitly, because the shifting property still holds in the multichannel case. Computation of the post-cursor ISI term is greatly simplified by using the following recursion:

$$\mathbf{x}_{pb}(n+1) =\downarrow \mathbf{x}_{pb}(n) + \hat{\mathbf{f}}_p(1,n)d(n),\ p = 1, \ldots P \tag{44}$$

where $\downarrow$ means shifting downwards by the oversampling factor (2, for a $T/2$ fractional spacing), and $\hat{\mathbf{f}}_p(1,n) =\downarrow \hat{\mathbf{f}}_p[n]$ is the shifted version of the current channel vector estimate. We note again that not all of the M elements of $\mathbf{x}_{pb}(n)$ are actually computed, but only those N that are needed for equalization.

In summary, the K input channels are pre-combined into $P \leq K$, which are used for channel estimation and equalization. A total of M signal samples per channel are used to obtain the M channel coefficients, but a number of those coefficients, say M_{off}, are set to zero when evaluating the post-cursor ISI. At the same time, only $N \leq M$ signal samples are used for equalization. The positions of the M_{off} taps are determined on-line, by thresholding. The positions of the N equalizer taps could theoretically be calculated on-line from the channel estimate; however, this calculation is extremely complex, and, thus, the N taps will be selected a priori in a practical implementation.

3.3 Algorithm summary

The algorithm for adaptive multichannel DFE, based on spatial pre-combining, channel estimation and sparsing, is defined by the following steps carried out at each iteration n:

(1) Form the matrix of input signals $\mathbf{V}(n)$. This matrix is of full size $K \times M$.

(2) Compute the pre-combiner output $\mathbf{X}(n) = \mathbf{C}'(n)\mathbf{V}(n)$. This matrix is of size $P \times M$.
($P \leq K$, i.e., size of spatial processing is reduced.)

(3) Compute the phase-corrected signals:

$$\mathbf{X}_\theta(n) = \begin{bmatrix} e^{-j\hat{\theta}_1(n)} & \cdots & 0 \\ 0 & \cdots & 0 \\ 0 & \cdots & e^{-j\hat{\theta}_P(n)} \end{bmatrix} \mathbf{X}(n) = \begin{bmatrix} \mathbf{x}_1^T(n)e^{-j\hat{\theta}_1(n)} \\ \vdots \\ \mathbf{x}_P^T(n)e^{-j\hat{\theta}_P(n)} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{1\theta}^T(n) \\ \vdots \\ \mathbf{x}_{P\theta}^T(n) \end{bmatrix} \tag{45}$$

This matrix is of size $P \times M$.

(4) According to a tap selection rule for equalization, take N elements from each row of $\mathbf{X}_\theta(n)$, and arrange them in a matrix $\mathbf{X}_\theta^{(s)}(n)$ of size $P \times N$.
($N \leq M$, i.e., size of temporal processing is reduced.)

(5) Compute $\mathbf{X}_b^{(s)}(n)$ by selecting the elements of $\mathbf{X}_b(n)$, the matrix whose rows are the post-cursor ISI terms $\mathbf{x}_{pb}^T(n)$.

(6) Compute $\mathbf{X}_f^{(s)}(n) = \mathbf{X}_\theta^{(s)}(n) - \mathbf{X}_b^{(s)}(n)$, the input signals to the feedforwad equalizer filters.

(7) Form the $K \times N$ matrix $\mathbf{V}^{(s)}(n)$ of selected signal components, and determine the equivalent pre-combiner input

$$\mathbf{y}(n) = \begin{bmatrix} \mathbf{V}^{(s)}(n)\mathbf{a}_1^{(s)*}(n)e^{-j\hat{\theta}_1(n)} \\ \vdots \\ \mathbf{V}^{(s)}(n)\mathbf{a}_P^{(s)*}(n)e^{-j\hat{\theta}_P(n)} \end{bmatrix} \tag{46}$$

(8) Compute $\alpha_p(n) = \mathbf{a}_p^{(s)\prime}(n)\mathbf{x}_{p\theta}^{(s)}(n)$, and $\alpha_{pb}(n) = \mathbf{a}_p^{(s)\prime}(n)\mathbf{x}_{pb}^{(s)}(n)$. The quantities with superscript (s) are of length N.

(9) Compute the data symbol estimate as in (37).

(10) After training ($n > N_t$), compute the data symbol decision $\tilde{d}(n)$ from the estimate $\hat{d}(n)$, using the decision regions appropriate to the modulation method used. Any linear modulation applies.

(11) Compute the error as in (31).
(12) Update the phase estimates as in (41).
(13) Update the pre-combiner weight vector

$$\mathbf{c}(n+1) = \mathbf{c}(n) + A_1[\mathbf{y}(n), e(n)] \tag{47}$$

(14) Update the equalizer vectors

$$\mathbf{a}^{(s)}(n+1) = \mathbf{a}^{(s)}(n) + A_2[\mathbf{x}_f^{(s)}(n), e(n)] \tag{48}$$

where $\mathbf{x}_f^{(s)}(n)$ is formed by arranging the rows of $\mathbf{X}_f^{(s)}(n)$ into a vector.
(15) Update the matrix of channel estimates

$$\mathbf{F}[n] = \lambda_{ch}\mathbf{F}[n-1] + (1-\lambda_{ch})\mathbf{X}_\theta(n)d^*(n) \tag{49}$$

(16) Truncate the channel estimates

$$\mathbf{F}^{(t)}[n] = \mathbf{F}[n]|_{\text{elements less than threshold set to 0}} \tag{50}$$

(17) Compute the post-cursor ISI term

$$\mathbf{X}_b^T(n+1) = \downarrow \mathbf{X}_b^T(n) + \downarrow \mathbf{F}^{(t)T}[n]\tilde{d}^*(n) \tag{51}$$

Initially, the values of the equalizers, the channel estimates, the phase estimates and the post-cursor ISI terms are all set to zero, and the pre-combiner weights are initialized such that arbitrarily chosen P input channels are passed to the equalizers unchanged, while the remaining channels are cut off. Updating of the pre-combiner is delayed with respect to the equalizer by a certain number of iterations N_{tc}. In the examples to follow, both the pre-combiner and the equalizers use the form II RLS [12]. This algorithm is of quadratic complexity; however, the two algorithms A_1 and A_2 are run in parallel, which increases the overall speed of computations. At the same time, it allows the pre-combiner and the equalizer to use different forgetting factors.

4 Experimental Results

The algorithm described in the previous section was implemented in Matlab and used for off-line processing of experimental data. The experimental data, provided by the Woods Hole Oceanographic Institution, were collected in the Continental Shelf region near the coast of New England. The water depth was between 100 m and 200 m. The signals were transmitted using a carrier frequency of 25 kHz, over the range of 3 km. The modulation format was QPSK, and the signals were transmitted at 10 kilobits per second. The vertical receiver array consisted of eight omni-directional hydrophones, spaced by

0.03 m. In this section, results of signal processing are demonstrated using an average quality data set.

The channel responses, recorded at four of the input sensors, are shown in Fig.5. These responses represent snapshots obtained from the channel probe transmitted before the actual data. They can be used to determine roughly the extent of multipath; however, they do not provide information about the time-variability of the channel. The channel response consists of the principal cluster of arrivals whose delay spread is on the order of 15 symbol intervals. In addition, there is a distant cluster of arrivals, approximately 225 symbol intervals away (the distant cluster is quite apparent in the response of the second sensor). Within each cluster, there are several distinct peaks of the response magnitude. The relative energy of multipath arrivals varies with the sensor location.

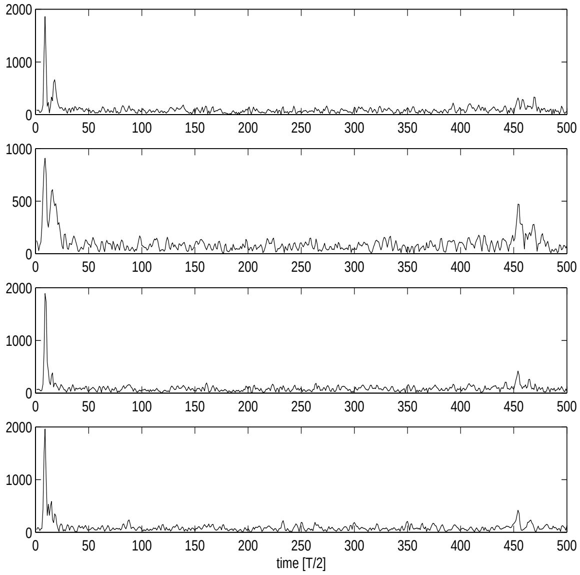

Fig. 5. Recorded channel responses.

As a reference, Fig.6 shows the results of signal processing using a conventional multichannel equalizer in a full-complexity configuration. The input to the receiver is the raw received signal, brought to baseband using the nominal carrier frequency, sampled at $2/T$, and frame-synchronized using a 13 element Barker code. No phase synchronization or bit-timing adjustment is performed on this signal. The receiver uses eight feedforward filters, each of size 16. The feedback filter is implemented explicitly, using 25 taps. The total of $8 \cdot 16 + 25 = 153$ equalizer parameters are updated as a single vector using the a numerically stable fast RLS [13] with a forgetting factor 0.99. There are 2000 symbols in a data packet, 300 of which are used for training. A single phase estimate is used for all eight channels. The details of tracking parameters are given in the figure. The performance of this receiver is of moderate quality, as seen by the output scatter plot and the estimated error probability (fraction of erroneous bits in the data packet). Note that if it were desired to extend the feedback section to take into account more distant arrivals, the size of the feedback filter would become a limiting factor in the efficiency of the algorithm that updates all the equalizer coefficients as a single parameter vector.

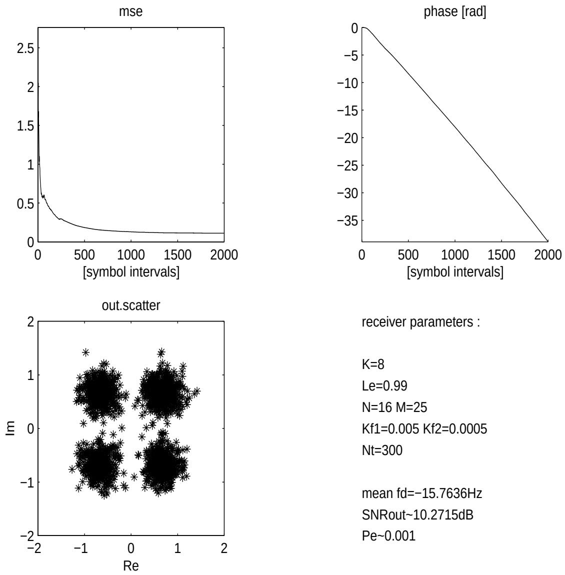

Fig. 6. Results of full-complexity multichannel processing. DFE is implemented explicitly. Figure shows the MSE, the phase estimate and the output scatter plot of the estimated data symbols. Equalizer coefficients are updated using the fast RLS algorithm [8].

In the absence of noise and time-variability of the channel, the performance of the full complexity receiver represents a bound on the performance of the reduced-complexity receiver. However, in a practical implementation this may not be the case. Fig.7 shows the results of data processing using the reduced-complexity channel-estimation-based DFE. The eight input channels are pre-combined into two channels, which are then equalized, i.e., processed by the two channel estimators that accompany the two feedforward filters. The channels are estimated over a total length of $M = 25$ taps. The truncation threshold for channel estimate sparsing is set to $G_0 = 1/6$, i.e., all the channel taps whose magnitude falls below 1/6 of the strongest tap magnitude (computed individually for each of the two estimators) are set to zero. This procedure results in $M_{off} = 18$ taps being turned off in each of the channel estimators. The remaining 7 taps are used in post-cursor ISI cancelation. The feedforward filters have $N = 17$ taps per channel. The forgetting factors of the equalizer, the pre-combiner and the channel estimators are indicated in the figure together with the PLL tracking constants. The pre-combiner (16 taps) and the two-channel equalizer (34 taps) each use a standard RLS algorithm, while the channel estimators are updated using the algorithm (43). Adaptation is carried out continuously throughout the data packet. The two-channel PLL tracks a Doppler shift of approximately -15 Hz with good accuracy. The MSE indicates steady convergence, and the scatter plot shows no errors in the detection of the data block.

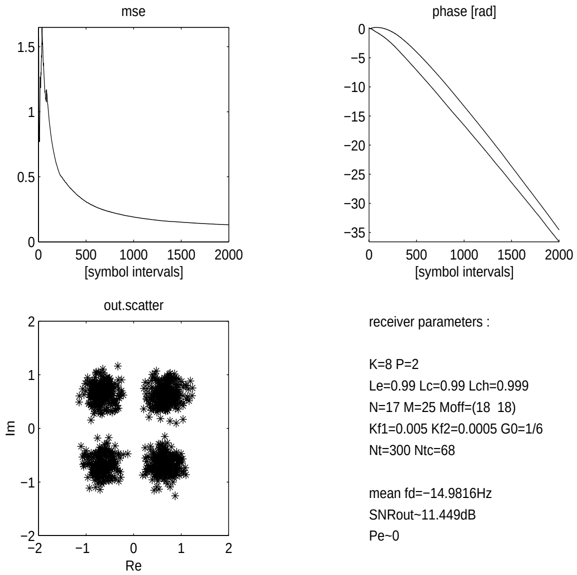

Fig. 7. Results of reduced-complexity multichannel processing. DFE is implemented via channel estimation. Figure shows the MSE, the phase estimates and the output scatter plot of the estimated data symbols. The pre-combiner and the equalizer coefficients are updated using standard RLS algorithms; the channel estimates are updated using the algorithm (43).

Increasing the number of equalizers (pre-combiner outputs) to more than two does not result in performance improvement on this channel. This fact illustrates the power of reduced-complexity multichannel combining. The pre-combiner succeeds in suppressing the distant multipath arrivals (more than 200 symbols away), as evidenced by the fact that only a short feedback is needed for suppression of ISI from within the first cluster. The small number of feedforward coefficients allows efficient processing without unnecessary noise enhancement.

The channel estimates obtained during data processing are shown in Fig.8. Recall that these estimates represent the channel responses as seen *after* pre-combining. Shown in the figure are both the complete channel estimates and their values after truncation. The ISI extends over 10 to 15 symbols. The channels exhibit a fair degree of time-variability. The variation occurs in both the amplitudes and arrival times of signals propagating over different paths. The latter is evident from the variation in position of the selected channel coefficients: the position of the strongest coefficient changes from the reference value 0 at the beginning of the data block to a lag of 4 (2 symbol intervals) at the end of the data block. For this reason, all the coefficients in the channel estimates need to be continuously monitored and updated. The adaptive fractionally spaced equalizers can then successfully compensate for the motion-induced drifting of the channel taps, as long as the resulting Doppler spread is relatively low.

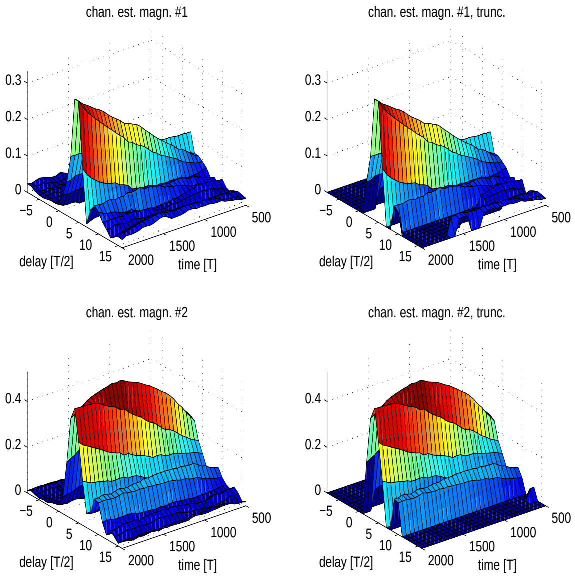

Fig. 8. Channel estimates after pre-combining. Shown on the left are the full channel estimates. Shown on the right are their truncated values which contain only the significant coefficients used for equalization.

As it was pointed out earlier, one advantage of the channel-estimation-aided DFE is that the channel estimator operates separately from the rest of the receiver, and thus provides the possibility to monitor the channel response without affecting the equalizer size. This point is illustrated in Fig.9, which shows the results of processing the same data packet with the span of channel estimators extended to include the distant multipath arrivals. The receiver performance in this case is again very good, with no detection errors. (The increase in the MSE occurs at the time when the late arrival becomes visible to the equalizer.) The channel estimators capture the distant arrivals, 225 symbols away from the main arrival. This example demonstrates a powerful advantage of sparse channel estimation: the overall performance is not affected by the large extent of the feedback, because only the significant taps are used for ISI suppression. It is thus possible to constantly monitor the channel changes without affecting the equalizer size (the feedforward filters in this example have 9 taps only). If the feedback section of a conventional DFE were extended to cover 225 post-cursors, this would significantly slow the convergence, and also restrict the tracking speed (by constraining the choice of the LMS step size or the fast RLS forgetting factor). Because different channel taps are updated independently in the algorithm proposed, convergence remains fast, channel estimation is not overly sensitive to the choice of forgetting factor λ_{ch}, and this factor is not restricted by the estimator size.

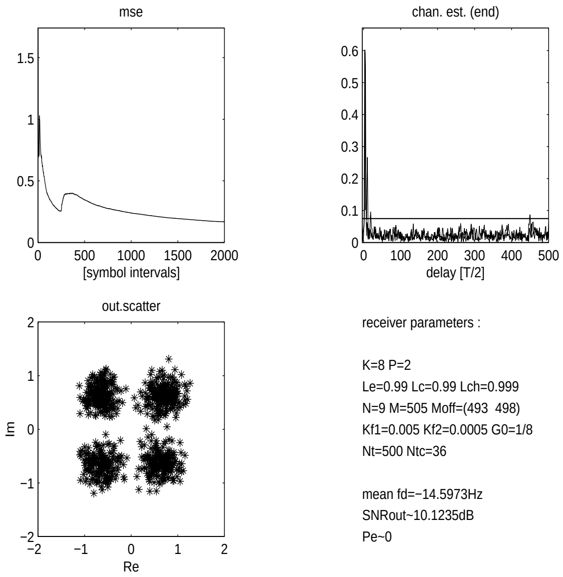

Fig. 9. Results of reduced-complexity multichannel processing. DFE is implemented via channel estimation. Figure shows the estimated mean squared error, the channel estimates and truncation thresholds (approximately same for the two estimates) at the end of the data packet, and the output scatter plot. In this case, the distant arrivals are taken into account.

5 Conclusions

Reduction of computational complexity is a problem of paramount importance for real-time implementation of high-speed underwater acoustic receivers which require sophisticated multichannel equalizers. This problem was addressed from the viewpoint of reducing the size of adaptive filters needed to perform spatial and temporal signal processing. In particular, the method of channel-estimation-aided decision-feedback equalization was used in conjunction with multichannel spatial diversity pre-combining. By reducing the number of channels that are equalized, and by selecting only the significant components of the channel estimates, not only is the efficiency of signal processing increased, but its performance is improved by eliminating the unnecessary noise from the detection process.

The ideas that govern the receiver implementation are:

(1) Pre-combining of a larger number of input channels into a few that will be equalized.
(2) Adaptive channel estimation and sparsing.
(3) Subtraction of post-cursor ISI using the channel estimate feedback prior to adaptive feedforward equalization.

The advantages of this method are the following:

(1) Preservation of full multichannel processing gain without temporal processing in each channel.
(2) Optimal selection of the channel taps by simple truncation in magnitude (rather than direct selection of the equalizer taps).
(3) Efficient post-cursor ISI calculation using the shifting property (44).
(4) Use of a short feedforward equalizer (possibly much shorter than the multipath spread).
(5) Computationally efficient estimation of the fractionally-spaced channel response using the recursion (43).
(6) Continuous channel monitoring.
(7) Flexibility to increase the feedback length on demand without affecting feedforward equalization.
(8) Parallel implementation of adaptive algorithms for channel estimation, equalization and pre-combining.

The algorithm that incorporates the above principles, together with a method for data-directed phase tracking, was successfully demonstrated on real data.

Further research in this area will likely concentrate on highly-mobile underwater acoustic communications, with relative transmitter/receiver velocities on the order of tens of meters per second. Because the speed of acoustic signal propagation underwater (1500 m/s) is very low, Doppler effects become a major limitation in mobile underwater channels. Signal processing based on channel estimation provides a natural framework for the development of algorithms capable of dealing with extreme motion-induced signal distortions.

References

[1] M.Stojanovic, J.Catipovic and J.Proakis, Adaptive multichannel combining and equalization for underwater acoustic communications, Journal of the Acoustical Society of America, vol.94 (3), Pt.1, pp.1621-1631, Sept. 1993.

[2] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski and K. Ball, The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms, in Proc. IEEE Oceans Conf., 2005.

[3] M.Stojanovic, J.Catipovic and J.Proakis, Reduced-complexity multichannel processing of underwater acoustic communication signals, Journal of the Acoustical Society of America, vol.98 (2), Pt. 1, pp.961-972, Aug. 1995.

[4] H.C.Song, W.Hodgkiss, W.Kuperman, M.Stevenson and T.Akal, Improvement of time-reversal communications using adaptive channel equalizers, IEEE J.Oceanic Eng., vol.31, No.2, April 2006, pp.487-496.

[5] M.Johnson, L.Freitag and M.Stojanovic, Efficient equalizer update algorithms for acoustic communication channels of varying complexity, in Proc. IEEE

Oceans Conf., 1997.

[6] M.Stojanovic, L.Freitag and M.Johnson, Channel-estimation-based adaptive equalization of underwater acoustic signals, in Proc. IEEE Oceans Conf., Sept. 1999.

[7] W.Li and J.Preisig, “Estimation of rapidly time-varying sparse channels,” *IEEE J.Oceanic Eng.*, in press.

[8] S.Ariyavisitakul and L.Greenstein, Reduced-complexity equalization techniques for broad-band wireless channels, IEEE Trans. Commun., pp.5-15, Jan. 1997.

[9] S.Ariyavisitakul, N.Sollenberger and L.Greenstein, Tap-selectable decision feedback equalization, IEEE Trans. Commun., pp.1497-1500, Dec. 1997.

[10] I.Fevrier, S.Gelfand and M.Fitz, ‘Reduced complexity decision feedback equalization for multipath channels with large delay spreads, IEEE Trans. Commun., pp.927-937, June 1999.

[11] C.Carbonelli, S.Vedantam and U.Mitra, Sparse channel estimation with zero tap detection, IEEE Trans. Wireless Commun., vol.6, No.5, May 2007, pp.1743-1753.

[12] S.Haykin, Adaptive Filter Theory, second ed., Prentice Hall, 1991.

[13] D.Slock and T.Kailath, Numerically stable fast transversal filters for recursive least squares adaptive filtering, IEEE Trans. Sig. Proc., pp.92-114, Jan. 1991.

List of Figures